ELECTIVE NO.1

Data Structures & Algorithms

Designed for Programmers and Developers

Data Structures & Algorithms

✓✓ Software architecture design, Data Structures and Algorithms are not only crucial to land your dream job, but it also helps in improving problem-solving and analytical skills that can prompt you to think out-of-the-box solutions.

✓✓ Master critical software design skills and crack product based MNCs and stay at the forefront of your future work.

✓✓ Best way to learn is to understand real-world problems and bugs encountered by organizations. Fix real-world coding/design issues guided by experts.

✓✓ Work on the most recent interview questions from Google, Microsoft, Amazon, and other comparable businesses, as well as issues from competitive platforms such as Leetcode, Hackerearth, HackerRank, etc.

Learn Advanced Problem Solving Technique

Crack coding interviews of product-based companies
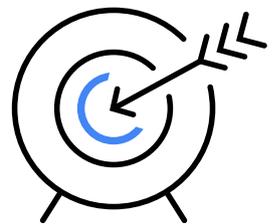
Work with real time projects through-out the elective

Data Structures & Algorithms

The fields of data science and artificial intelligence use a wide range of approaches, including statistical analysis, modelling, machine learning, and data mining, to help us forecast the future.

Who should join?

- Software developers/Programmers, Technical Architect , Automation Test Engineer, Java and .NET Developer.

- Python, Embedded developer, Android/iOS developer.

- Professionals planning to crack interview in top product based companies as AI expert.

Why data structures & Algorithm?

- Data structures and Algorithms has become important when you want to learn in-depth and complex AI and ML algorithms.

- Data structures and Algorithms helps you to crack interview in product based MNCs as ML and AI expert.

- Even so, these are the main reasons why experienced workers seeking career changes are in greater demand.

Course Pre - requisite:
1+ year of experience in coding/programming.

Tools & Modules

## TERM 1 & 2

Python

Statistics

Machine Learning

Deep Learning (Tensorflow)

Time Series Analysis & Forecasting
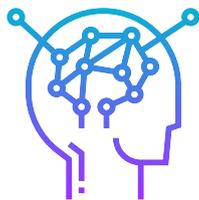
Natural Language Processing
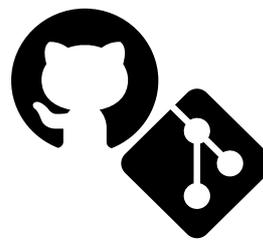
Git & GitHub

R Programming

TERM 3 & 4

**Reinforcement Learning**

Open CV

Pytorch

Keras

Google Cloud
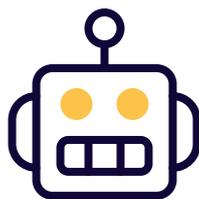
Advance AI

Project Expertise

Elective - DSA

Program RoadMap

100% Interview Guarantee in Product based MNCs

5

4

Job Preparation
(Resume Build- Up, Mock Interview, Job Referrals)

Work on Real Time Projects and learn DSA with 100+ coding interview questions.

3

Analyze your knowledge and interest towards any 2 domain from Domain Electives

2

1

Complete General Program (Term 1 to 4) Core + Advance Modules & Tools

Main Brochure

What will you learn :

Data Structures & Algorithms

Master Important Algorithms:

- Analysis of algorithm
- Searching algorithm
- Sorting algorithm
- Recursion algorithm
- Greddy algorithm

- Dynamic Programming
- Backtracking algorithm
- Pattern Searching
- Divide and Conquer
- Geometric algorithm
- Graph algorithm
- BIT algorithm

Mastering - Data structure and Algorithms with 100+ coding interview questions.

DSA Training

40 hrs | 6 weeks

Module 1 - Array and Array ADT

What is an array? will be discussed in this segment. What method is used to store it in memory? What is the difference between a static and a dynamic array? How can the size of an array be increased?

Module 2 - Strings

In this section, we will look at how to find the length of a string, change the case of a string, validate a string, reverse a string, count words and vowels in a string, compare strings and find duplicates in a string in a normal way, as well as using bitwise operations and checking whether two strings are anagram? We will also discuss Rabin Karp and KMP algorithms here.

Module 3 - Linked List

This segment covers the following topics: What exactly is a linked list and why do we need it? What is the definition of a singly connected list? What is a Doubly Linked List, and how does it work? What is a Circular Connected List, and how does it work? What is a Circular Doubly Linked List, and how does it work?

Module 4 - Stack

This segment covers the following topics: What is stack? What is the difference between LIFO and FIFO principles? What is the role of the stack? Push(), pop(), isempty(), isfull(), peek(), count(), change(), display(), and other typical stack operations. We'll even go over a few real-world stack use cases.

Module 5 - Queue

In this part, we'll look at what a Queue data structure is and how it functions. Real-life examples of the same. Linear queues, circular queues, priority queues, and deque queues are examples of queue types. We'll also hear about Enqueue, Dequeue, Peek, Queue Overflow, and Queue Underflow, as well as other queue operations.

Module 6 - Tree, Binary Search Tree and AVL Tree

We'll learn about Tree Data Structure and terms like Root, Child node, Parent, Sibling, Leaf node, Internal nodes, Ancestor nodes, and Descendent in this section. Implementation of Tree and Tree Traversal (such as Inorder, Preorder, Postorder, and so on) Segment Tree also we will cover in this section.
The following topics will be discussed: BST implementation of search, insertion, deletion, and floor, self-balancing BST, Tree set and Tree map, depth and height of nodes, Level order traversal as well as BST application.

Module 7 - Hashing

What is a Hash table? would be covered in Hashing. What exactly is hashing? There are many methods for calculating hash functions, including the Division, Folding, and Mid-square methods.

DSA Training

40 hrs | 6 weeks

Module 8 - Heap

This section will introduce us to Heap Data Structure and its implementation. We will also learn about binary heap and various operations like Insertion, Heapify and extract, Decrease key, Delete and Build map.

Module 9 - Trie

We will learn about what Trie is? Properties of trie for a set of string, searching, inserting and deleting a node from Trie, Application, Advantages & Disadvantages of a Trie and Counting distinct rows in a binary matrix.

Module 10 - Graph & Recursion

Graph representation, BFS, DFS, Shortest path in Directed Acyclic graph, Prim's algorithm and minimum spanning tree, Dijkestra's shortest path algorithm, Kruskal's algorithm, Kosaraju's algorithm, Articulation point, Bridges in graph, Tarjan's algorithm.

Introduction to recursion, Application to recursion, writing base cases and problems discussed here are kind of Tower of Hanoi, Josephus problem

Module 11 - Searching & Sorting

Linear search, binary search, BFS, DFS, Two pointer approach problem, Ternary search, Jump search, Exponential search, and other topics will be addressed.

In sorting we will discuss Bubble sort, Bucket sort, Comb sort, Counting sort, Heap Sort, Insertion sort, Merge sort, Quick sort, Radix sort, Selection sort, Shell sort, Bitcoin sort, Cocktail sort, Cycle sort, Tim sort.

Module 12 - Greedy Algorithm

The topics for discussion over here are the Activity selection problem, Fractional Kanpsack, Kruskal's minimum spanning tree problem, Huffman coding, Prim's MST for Adjacency List Representation, Greedy Algorithm to find minimum number of Coins etc.

Module 13 - Backtracking Algorithm

- Rat in a maze problem
- Knight's tour problem
- Hamiltonian cycle
- Tug of war

Module 14 - Pattern searching

- Naive pattern searching
- KMP algorithm
- Finite automata
- Boyer Moore algorithm

Module 15 - Dynamic Programming

- Edit distance problem using naive and DP approach
- 0-1 Knapsack problem using naive and DP approach
- Optimal strategy for a game
- Egg dropping problem
- Coin change problem

FAQ's

+ Can I select multiple domain electives?

▬ You can select multiple electives based on your career goal and work experience/academics.

+ What if I don't have any prior experience in any domain?

▬ Even if you don't have any prior experience, you can still opt for any elective to gain Domain Expertise and work on Real - Time Industrial Projects.

+ Can I change my domain electives later ?

▬ Yes, you can change your elective or repeat the training later within the Course Accessibility Duration.

+ Are there any additional charges for electives?

▬ No, there are no additional/ hidden charges.

+ How many capstone projects do I need to work?

▬ You can work on all projects, or depending on your experience and goal. For eg, Having 1-2 yrs of experience you must work on 4-5 projects.

Note: We keep updating trending projects and case - studies as per the market/company requirement. You can also Bring your own project.

Linked in | **Stay updated with newest content (Infographics, Interview Q&A, Job Updates and more) on Data Science and AI.**

YouTube | **Subscribe to our YouTube Channel and Watch Full Tutorial of Domain Specific Projects, Guided by Industrial Experts**

STILL CONFUSED?

Apply for **FREE** Career Counselling Session with our Expert

📞 **+91 77 956 87 988**

BOOK NOW